System Specification

System Integrity Thread Assessment

Thor DP-1

Checkout and Launch Control System (CLCS)

84K00302-001
Version 1.8

Approval:

Date	Date
Date	Date
Date	Date

NOTE: See “**Supporting Document Note**” on following page

PREPARED BY: Ken Clark

Assessment Team

Name	CI Represented	Phone
Aiello, Pete	O&M User	1-7294
Blackledge, Jack	Systems Engineering	1-2264
Bullington, Joseph	LDB GW SSI	1-4077
Chen, Cecelia	Data Distribution & Processing	1-2298
Clark, Ken	SE Thread Lead	1-2262
Cummins, Scott	LDB GW SSI	1-4654
Dawson, Rick	Systems Engineering	1-2296
Duncan, Carl	O&M User	1-6124
Foster, Earl	SDC SSI	1-7710
Hrezo, Gary	System Integrity Viewer	1-7445
Jamieson, Tom	Systems Engineering	1-2263
King, Charla	Test Build CSCIs	1-7587
Le, Chau	PCM GW SSI	1-2293
Lunceford, Mike	GSE GW SSI	1-7557
McMahon, Bob	System Services	1-7396
Moore, Steve	C&C W/S SSI System Integrity CCP SSI DDP SSI Ops CM/Activity Manager	1-7394
Morales, Alex	Network Services	1-7444
Porter, John	Network Services	1-7311
Quinn, Shawn	Gateway SSI	1-7608
Raucci, Jack	System Viewer	1-7493
Samson, Julia	Application Services	1-2213
Wilkinson, John	Systems Engineering	1-7390

Supporting Document Note: Acronyms and definitions of many common CLCS terms may be found in the following documents: CLCS Acronyms 84K00240 and CLCS Project Glossary 84K00250.

REVISION HISTORY

REV	DESCRIPTION	DATE

LIST OF EFFECTIVE PAGES				
Dates of issue of change pages are:				
Page No.	**A or D***	**Issue or Change No.**	**CR No.**	**Effective Date****

Table of Contents

SYSTEM INTEGRITY THREAD ASSESSMENT THOR DP 1

1. INTRODUCTION

1.1 SYSTEM INTEGRITY THREAD OVERVIEW.

This thread provides the supporting System and Subsystem Integrity infrastructure as a foundation for implementing Redundancy Management in Atlas. The focus in Thor will be in defining, transmitting/logging, and displaying Subsystem health and performance information. The beginnings of Set configurability will be implemented (i.e., A System Configuration Table (SCT) will be defined and implemented). Modes of operation for all Subsystems will be designed and implemented for Thor. The top level design for Redundancy Management will be defined, but not implemented in Thor.

1.2 SYSTEM INTEGRITY THREAD CONCEPT

System Visibility and Redundancy Management in an RTPS Test Set is accomplished by a set of programs executing in each subsystem in the Test Set. Subsystem Integrity exists in every computer in the Test Set and reports subsystem health, status and activity within the local subsystem to System Integrity which executes in a CCP. In addition to the Subsystem/System Integrity programs, there is a viewer that executes in any Command and Control Workstation. This System Status viewer can view the overall status of the Test Set or the detailed status of any subsystem in the Test Set. Figure 1 — System/Subsystem Integrity Topologybelow shows the allocation of System and Subsystem Integrity in a typical RTPS Test Set. **Note:** No redundancy is implemented in the Thor delivery.

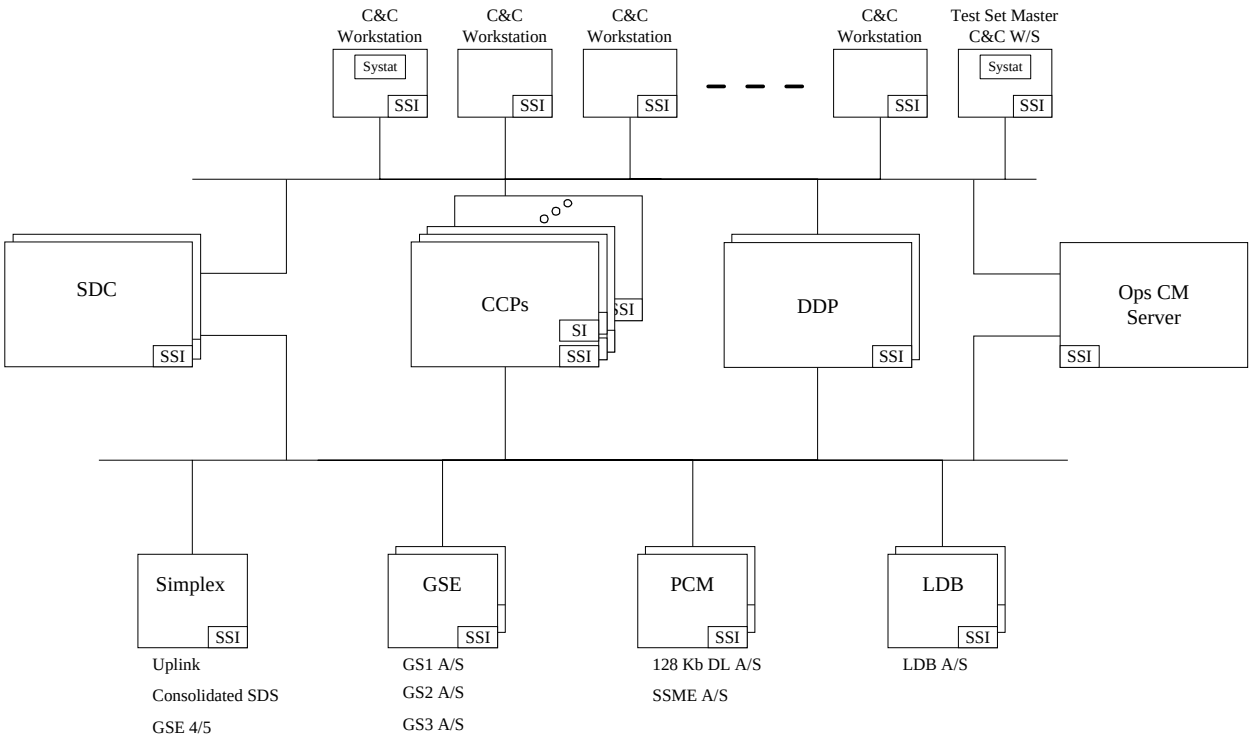

Figure 1 — System/Subsystem Integrity Topology

Note: The Uplink, Consolidated SDS, and GSE 4/5 Gateways are shown here as Simplex systems, some will be designed to operate as redundant systems. The decision to run a Subsystem as a simplex or redundant system will be an operational one made by O&M personnel based on test requirements for availability of the data stream and

the availability of hardware to execute in a redundant manner.

1.2.1 Concept of operations

RTPS Set Visibility and Redundancy Management is accomplished by a set of programs that execute in the subsystems of the set. These programs known as “Subsystem Integrity” exist in each subsystem. Subsystem Integrity is responsible for:

1. Monitoring Subsystem health and status
2. Communicating subsystem health
3. Communicating status and performance information
4. Maintaining the System Configuration Table as directed by Ops CM and System Integrity

Subsystem Integrity in each of the Subsystems monitors the health of the platform and all applications on the platform (See Section 1.2.2 for more details). Applications will be defined as critical or non-critical. If the health of a critical application is “no go”, the fact is reported to Subsystem Integrity and a Critical Process No Go will be communicated to System Integrity. In the Thor release Critical Applications will only be those that reside in gateways and are those that are required to make the gateway perform it’s function.

A set of System Status FDs will be defined for all subsystems and subsystem devices. Subsystem Integrity in each of the subsystems will create and maintain the data required by the FDs and introduce these FDs as System Status FDs at the appropriate rate into the data stream as depicted in Figure 2 — Generic Subsystem & Subsystem Integrity. Data Distribution processes the System Status FDs as it would any other FD. Table 1 — Active/Standby Subsystem Transmission Rate contains the initial estimates for transmission rates for Subsystem Health Counter and Subsystem Status FDs.

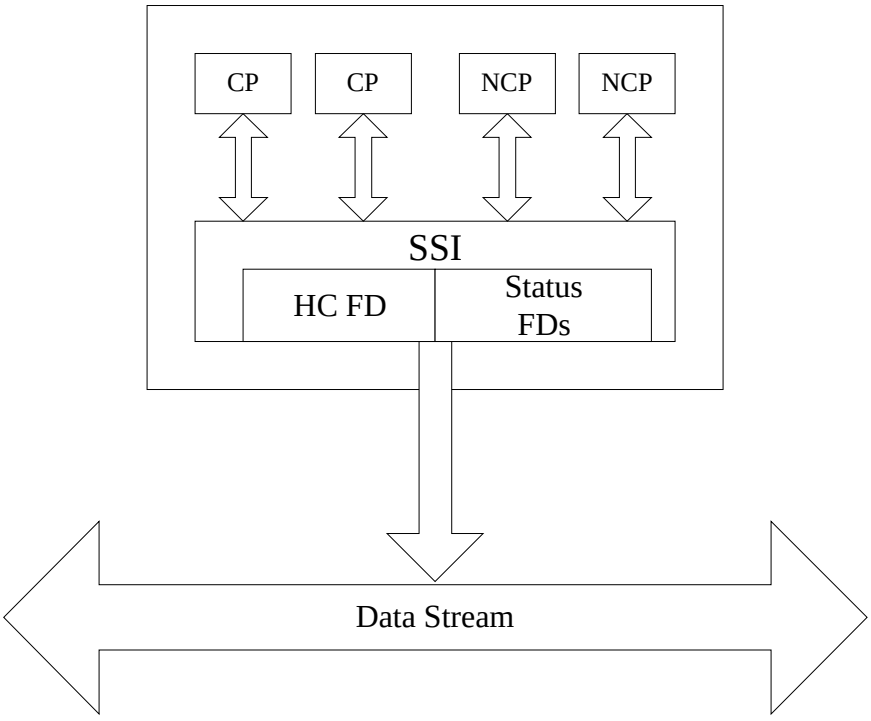

Figure 2 — Generic Subsystem & Subsystem Integrity

Definitions

Critical Process — A software or hardware process in a subsystem that must be operating properly for the subsystem to be in the “GO Mode”

Non Critical Process — A software or hardware process in a subsystem that is not required to be operating properly for the subsystem to be in the “GO Mode”

Subsystem — The collection of hardware and software that is combined to perform a specific set of functions (e.g.

GSE Gateway, CCP, DDP).

Subsystem Health FD — The single FD from each subsystem that indicates the subsystem is performing its function.

Subsystem Status FDs — The set of FDs that provide detailed status of the subsystem (e.g., Use & Error Counters, Format IDs being processed, etc.)

Subsystem Integrity in each of the RTPS Subsystems introduces Subsystem Health FDs and Subsystem Status FDs (e.g., Health Counters, Use & Error Counters, etc.) as follows:

	Active Subsys Xmit Rate		Standby Subsys Xmit Rate	
Subsystem	HC FD	Status FDs	HC FD	Status FDs
GSE G/W	SSR	P/C/D	1/Sec	P/C/D
PCM DL G/W	SSR	P/C/D	1/Sec	P/C/D
SSME GW	SSR	P/C/D	1/Sec	P/C/D
LDB G/W	SSR	P/C/D	1/Sec	P/C/D
PCM UPLK G/W	SSR	P/C/D	1/Sec	P/C/D
Consolidated G/W	SSR	P/C/D	1/Sec	P/C/D
DDP	SSR	P/C/D	1/Sec	P/C/D
CCPs	SSR	P/C/D	1/Sec	P/C/D
Ops CM Server	1/Sec	P/C/D	1/Sec	P/C/D
SDC	1/Sec	P/C/D	1/Sec	P/C/D
C & C W/Ss	1/Sec	P/C/D	1/Sec	P/C/D

Table 1 — Active/Standby Subsystem Transmission Rate

P/C/D => P = Periodically — At the rate defined for the FDs. E = When an error occurs in a Subsystem, all System Status (i.e., use, error, and performance) FDs for that Subsystem will be introduced into the data stream at the next update cycle for the HC FD (Change). D = A capability to update the error counts on demand must also be provided (Demand).

Standby GSE Gateway monitors the active GSE Data Bus to determine if the active Gateway is still performing its function. During this period it collects measurement data, and is prepared to send a backup change packet if requested by Data Distribution. If the Standby Gateway sees no activity on the GSE Data Bus, it notifies System Integrity of "No Bus Activity". GSE Standby Gateway also receives commands from the CCP so that it tracks the Active Gateway and is prepared to issue any commands not issued by the Active Gateway if a switchover is required and directed by System Integrity.

Standby LDB Gateways monitor the active LDB Gateway to determine if the active Gateway is still performing its function. During this period the Standby Gateway monitors GPC response data, and is prepared to send a backup response packet if requested. If the Standby Gateway sees no activity on the LDB it notifies System Integrity of "No Bus Activity". LDB Standby Gateway also receives commands from the CCP so that it tracks the Active Gateway and is prepared to issue any commands not issued by the Active Gateway if a switchover is required and directed by System Integrity.

Standby PCM Gateway monitors — TBS.

Data Distribution receives Data Change packets from Gateways containing change data, and subsystem health FDs. *If a Data Change packet is not received from a Gateway within TBD MSec after expected, the Data Change Packet is retrieved from the Standby Gateway and System Integrity is notified of the missed packet*

Both Active *and Standby* System Integrity receive the Subsystem Health FDs for all of the Subsystems in the Test Set *and analyze them to determine if any switchovers are needed* . If no changes are needed both Active *and Standby* System Integrity update their Current Status Information. If a health counter is missed Active System Integrity notes the fact. If the next expected health counter or data packet is received, Active System Integrity outputs a System Message. *If two successive Health FD updates or Data Change Packets are missed for a*

subsystem, Active System Integrity will cause a switchover to take effect under the following conditions:

1. *Subsystem is redundant.*
2. *Standby Subsystem is ready to assume active role.*
3. *Switchover is enabled for the redundant pair.*

Standby System Integrity monitors the health of Active System Integrity. If the health counter of the Active System Integrity is missed, Standby System Integrity notes the fact and continues. If the next expected health counter or data packet is received, Standby System Integrity outputs a System Message. If Active System Integrity fails to update its health counter FD for two consecutive times Standby System Integrity will order Active System Integrity to the standby state and become active.

1.2.2 Subsystem Monitoring

A set of non-intrusive subsystem checks will be defined for each subsystem and included as part of Subsystem Integrity. These tests will:

1. *Analyze the subsystem hardware and software*
2. *Detect errors in the hardware and take appropriate action*
   - *Report all critical errors to O&M*
   - *Tally non-critical error and report when they exceed a pre-established error threshold*
3. *Analyze the boot error log and report errors to O&M*
4. *Detect errors in the operational status of all critical and non-critical SW processes*
5. *Detect and report errors in the software configuration*

These subsystem tests will have the following characteristics:

1. *Run in the background*
2. *Default runtime from subsystem initialization until subsystem termination*
3. *The capability to start and stop by feature and feature group must exist.*

1.2.3 Simplex vs. Redundant

As may be seen in Figure 1.0 the RTPS is composed of simplex and redundant subsystems. If a failure occurs in a simplex subsystem the subsystem is marked as being in the "Not Go" mode and recovery operations will have to be accomplished. All subsystems in Thor will be simplex. **Post Thor** redundant subsystems will be introduced. When a failure occurs in a redundant subsystem a switchover is accomplished to the standby subsystem when the standby subsystem is available, ready to assume an operational mode, and switchover is enabled. Gateway Subsystems will be designed and developed to be redundant subsystems as described in **Table 2 Redundant Subsystem Implementation Matrix**. The decision to run a G/W subsystem as simplex or redundant is an operational decision based on availability of resources.

Subsystem	Redundant Implementation
CCP	Yes
DDP	Yes
C&C W/S	No
GSE Gateway	Yes
PCM Downlist Gateway	Yes
LDB Gateway	Yes
Uplink Gateway	Yes
Consolidated Systems Gateway	TBD
Ops CM Server	No
SDC Interface	TBD

Table 2 Redundant Subsystem Implementation Matrix

1.2.4 System Configuration Table

The System Configuration Table is created and may be modified by Ops CM for Thor. In later deliveries, the SCT will be created by a TBD CSC on the SDC. This TBD SDC CSC will have the task of creating SCTs for RTPS Test Sets. In later versions the TBD CSC will also define what resources are needed for different activities that will be performed in the Test Set as depicted in Table 3 — SCT Augmented by Activity Information. It is intended that the TBD CSC will create configurations that may be used in an RTPS Test Set without change or that Activity Management will have the ability to augment the configuration, or change it as required to support test operations. For Thor the SCT (see section 6.1 — System Configuration Table for details) will contain information such as:

1. Ref Des
2. Host Name
3. Subsystem Name
4. Subsystem Type
5. Attached To
6. Current State

For later deliveries the SCT will be augment with information such as:

	Activity ABC	Activity DEF		Activity XYZ
Subsystem A	Application Configuration HW Resources Required	Application Configuration HW Resources Required	•	Application Configuration HW Resources Required
Subsystem B	Application Configuration HW Resources Required	Application Configuration HW Resources Required	•	Application Configuration HW Resources Required
•	•	•		•
•	•	•		•
Subsystem N	Application Configuration HW Resources Required	Application Configuration HW Resources Required	•	Application Configuration HW Resources Required

Table 3 — SCT Augmented by Activity Information

Appropriate Applications and System Services APIs will be provided to access the data in the SCT.

1.2.5 Subsystem States

A consistent set of Subsystem states will be defined and implemented for Thor. Any subsystem on the Test Set can exist in one of these states. Section 5.2 contains the current design of the Subsystem State Matrix. The going in position to be refined as the design progresses is that there are four states as follows:

1. Subsystem in configuration — The named subsystem is in the configuration.
2. Subsystem TCID Loaded — The named subsystem has it's TCID information loaded
3. Subsystem communicating — The named subsystem is communicating.
4. Subsystem Go — The named subsystem is performing it's major function (e.g., GSE G/W acquiring data)

Note: Go mode for CCPs is totally different than for CCMS Consoles. In CCMS - Consoles go into go mode as soon as the Subsystem has been loaded. In CLCS - CCPs will have two kinds of operational "Go" mode states. Subsystem operational (O&M GO), and Applications operational (Subsystem Go). Exactly when and under what conditions these modes are achieved will be defined in Thor, but not implemented until Atlas or later.

1.2.6 System Event Codes

A set of System Event Codes will be defined. Most of these codes relate to the management of the System Configuration Table. Those that relate only to Redundancy Management will not be implemented for Thor. The proposed codes are as follows:

System Event	Xmitter	Rcvr	Event
Loaded	SSI SI	SI All SSI	Sent when the Subsystem has TCID Loaded Broadcast to all when SI receives TCID Loaded
Not Loaded	SSI SI	SI All SSI	Sent when the Subsystem transitions less than loaded Broadcast to all when TCID loaded is lost
Communicating	 SI	 All SSI	Recognized by SI when Health FD starts counting. Broadcast to all when SI detects subsystem communicating.
Not Communicating	 SI	 All SSI	Recognized by SI when Health FD stops counting. Broadcast to all when subsystem stops communicating.
Go	SSI SI	SI All SSI	Sent when the Subsystem enters Go mode Broadcast to all when SI receives subsystem Go
Not Go	SSI SI	SI All SSI	Sent when the Subsystem leaves Go mode for a lower mode Broadcast to all when subsystem leaves Go
HIM Status Change	GSE SSI SI	SI All SSI	Sent when HIM Status Change Broadcast to all when SI receives HIM status change.

1.3 SYSTEM INTEGRITY & SUBSYSTEM INITIALIZATION THREAD SPECIFICATION

1.3.1 Statement of Work

The SOW for the System Integrity Thread and Subsystem Initialization has been updated as shown with revision marking.

- Define standard formats for transmitting/logging Subsystem Integrity information, including: summary subsystem health counter, subsystem state, health counters of strategic processes, device errors, device usage counts, strategic performance statistics.
- From RTCN based subsystems provide Subsystem Integrity information to System Integrity at the System Synchronous Rate.
- From HCI subsystems provide Subsystem Integrity information to System Integrity.
- From the DDP provide Subsystem Integrity information to System Integrity.
- From the primary CCP provide aggregate Subsystem Integrity information every 1 second on the DCN to all HCI's
- Prototype the interface between Subsystem Integrity and User Applications on CCPs and CCW/Ss.
- Provide a System Configuration Table that defines the Subsystems that exist in the Test Set and that facilitates Redundancy Management post Thor.
- Define and implement System States for all Subsystems.
- Provide a Subsystem Health Viewer
    - Provide a system status HCI display that shows overall Subsystem Integrity status of all subsystems.
    - Provide a summary level display will show status of all subsystems on a single page.
    - Provide a single subsystem display, showing detail status of each subsystem.

- Both displays should be automatically refreshed at the aggregate as required.
- Provide a Subsystem Initialization CSC in each RTPS Subsystem
  - Define and Provide Real Time Process Priority for each delivered Thor RTPS CSC.
  - Utilize the SCT to establish subsystem identification for subsystems participating in the Test Set.

1.3.2 Requirements

(SLS - 2.2.9.1.5) The RTPS shall provide a set of visual displays that provide comprehensive insight into the state and configuration of the set resources (e.g., network resources, subsystem assignments, software configuration, etc.).

(SLS - 2.2.9.1.6) The RTPS shall provide different views of Test Sets and activities in configurable sets (e.g., Master Set View, Test Set View, Activity View).

(SLS - 2.2.9.2.8) The RTPS shall provide a visual display depicting the health and status of all hardware resources within a Test Set and within all Test Sets of a Configurable Set.

(SLS - 2.2.9.2.9) The RTPS shall provide the capability to monitor the configuration of each subsystem participating in a test including what software is executing and any subsystem error conditions.

(SLS - 2.2.9.2.10) The RTPS shall provide a central point for the display of system error, status, and mode change messages.

(SLS - 2.2.9.2.15) The RTPS shall provide the capability to continuously monitor subsystem resource utilization in all RTPS subsystems.

Redundancy Management

(SLS - 2.2.9.3.1) The RTPS shall provide redundancy management of all redundant subsystems and network resources in a Test Set.

(SLS - 2.2.9.3.2) The RTPS shall provide a central point to coordinate and direct redundant element activation (known as System Integrity).

(SLS - 2.2.9.3.3) System Integrity shall be capable of being run from any Console Position within a Test Set.

(SLS - 2.2.9.3.4) The RTPS shall provide the capability for a Standby copy of System Integrity to run and monitor the activities of the Active copy of System Integrity.

(SLS - 2.2.9.3.5) When the Standby copy of System Integrity determines that the Active copy is not operating properly it shall assume the role and responsibilities of the Active System Integrity.

(SLS - 2.2.9.3.6) The RTPS shall provide a method to share current configuration data with a redundant element.

(SLS - 2.2.9.3.7) The RTPS shall provide a method to track redundant element states.

(SLS - 2.2.9.3.8) System Integrity shall monitor critical subsystems for failure and in the event a monitored subsystem fails, shall perform a switchover (if enabled) to the standby subsystem.

(SLS - 2.2.9.3.9) System Integrity shall report all subsystem errors to a central point.

(SLS - 2.2.9.3.10) The CLCS shall provide a reduced capability mode in which a Test Set continues to support

even though all copies of System Integrity fail.

Note: Functions that are not supported or whose capability is reduced in the reduced capability mode are:

1. Redundant Element Switchover
2. Test Set Resource Monitoring
3. Checkpointing
4. Restarting subsystems

(SLS - 2.2.9.3.12) The CLCS shall provide a "warm boot" capability in which System Integrity can be restored after failure.

(SLS - 2.2.9.3.13) After a "warm boot" System Integrity shall restore normal function to those capabilities which were reduced while in the reduced capability mode.

1.3.3 System Control CSCI Description

CSCI	CSC	Function	Description
System Control CSCI			This CSCI provides the capability to control the RTPS.
	System Integrity		
		Redundancy Management	This CSC provides the capability to monitor Subsystem Integrity data, detect and handle redundancy issues. These services manage redundancy within RTPS for managing system loading, detecting failed subsystems and starting recovery actions.
		Subsystem Integrity	This CSC resides in every processor in the RTPS. It retrieves and transmits CMP and HMP programs and Subsystem health, and performance data to the Redundancy Management function.
	Ops Configuration Manager		This CSC provides the capability to configure and manage RTPS Test Resource Equipment in order to support RTPS operations.
		Activity Management	This CSC provides the capability to manage RTPS test activities. This includes allocation of Test Resource Equipment to an activity, managing activity state, and other OPS CM functions.
		RTPS System SW Load and Initialization	This function provides the capability to load the RTPS System Software into the Test Resource Equipment.
		Test Load	This function provides the capability to load the Test Build Software into the Test Resource Equipment.
		System & Test Load Verification	This function provides the capability to verify that the software loaded in the TRE is the software that is supposed to be loaded there. It also verifies that none of the required files have been corrupted.

Recommended Changes

1. Add System Status Viewer under System Viewers.
2. Add new CSC in SDC to build SCT and define resources required for activities (Planning and Resource Manager)
3. Change name of CSCs in the System Control CSCI as follows:

Option 1 — Selected by DP1 8/26/97

System Control CSCI
- Redundancy Management
  - System Integrity
  - (CPU Name) Subsystem Integrity
- Ops Configuration Management
  - Activity Management
  - RTPS System SW Load and Initialization
  - Test Load
  - System & Test Load Verification

Option 2

System Control CSCI
- System Integrity
- (CPU Name) Subsystem Integrity
- Ops Configuration Management
  - Activity Management
  - RTPS System SW Load and Initialization
  - Test Load
  - System & Test Load Verification

1.4 SYSTEM INTEGRITY THREAD HARDWARE DIAGRAM

Not Applicable.

1.5 SYSTEM INTEGRITY THREAD DELIVERABLES

Provide a list of deliverable products for this capability (e.g., CSCI Object files)

Deliverable	R&D Document	Code	API Manual	Users Guide
System Integrity	Y	Y		Y
Command & Control Workstation Subsystem Integrity	Y	Y		N/A
CCP Subsystem Integrity	Y	Y		N/A
DDP Subsystem Integrity	Y	Y		N/A
Ops CM Subsystem Integrity	Y	Y		N/A
Data Distribution & Processing	Y	Y		N/A
GSE Gateway SSI	Y	Y		N/A
PCM Gateway SSI	Y	Y		N/A
LDB Gateway SSI	Y	Y		N/A
System Viewers	Y	Y		Y
Application Services‘	Y	Y		N/A
Test Build CSCs	Y	Y		Y

1.6 SYSTEM INTEGRITY THREAD ASSESSMENT SUMMARY

1.6.1 Labor Assessments

The total Labor Costs required to provide this capability are summarized in the following table;

No.	CSCI/HWCI Name	Thor LM	Changes covered in
1	System Integrity	5.0	This Assessment
2	C&C W/S CSCs	4.0	This Assessment
3	CCP CSCs	4.0	This Assessment
4	DDP CSCs	4.0	This Assessment
5	Ops CM CSCs	4.0	This Assessment
6	Data Distribution & Processing	0	Data Distribution Completion
7	GSE Gateway CSCs	9.0	This Assessment
8	PCM Gateway CSCs	9.0	This Assessment
9	LDB Gateway CSCs	9.0	This Assessment
10	System Viewers	6.0	This Assessment
11	Application Services‘	3.0	This Assessment
12	Test Build CSCs	4.0	This Assessment
13	Integration & Test	TBD	This Assessment
	TOTAL	61.0	

1.6.2 Hardware Costs

None.

1.6.3 System Integrity Thread Procurement

None.

1.7

1.7.1 Labor Assessments

The total Labor Costs required to provide this capability are summarized in the following table;

No.	CSCI/HWCI Name	Thor LM	Changes covered in
1	System Integrity	5.0	This Assessment
2	C&C W/S CSCs	4.0	This Assessment
3	CCP CSCs	4.0	This Assessment
4	DDP CSCs	4.0	This Assessment
5	Ops CM CSCs	4.0	This Assessment
6	Data Distribution & Processing	0	Data Distribution Completion
7	GSE Gateway CSCs	9.0	This Assessment
8	PCM Gateway CSCs	9.0	This Assessment
9	LDB Gateway CSCs	9.0	This Assessment
10	System Viewers	6.0	This Assessment
11	Application Services‘	3.0	This Assessment
12	Test Build CSCs	4.0	This Assessment
13	Integration & Test	TBD	This Assessment
	TOTAL	61.0	

1.7.2 Hardware Costs

None.

1.7.3 System Integrity Thread Procurement

None.

1.8 SYSTEM INTEGRITY THREAD SCHEDULE & DEPENDENCIES

1.8.1 Schedule

Task Name	Start	Finish
Thor Assessment Kickoff	07/25/97	09/30/97
Concept Panel Internal Review	N/A	09/30/97
Concept Panel	N/A	10/02/97
Thor Development		
Requirement Panel Internal Review	N/A	10/28/97
Requirement Panel	N/A	10/30/97
Design Panel Internal Review	N/A	12/02/97
Design Panel	N/A	12/04/97
CSCI Unit Testing	01/15/98	01/30/98
CSCI Development Integration Test	02/02/98	02/13/98
CSCI Formal Integration Test	02/18/98	02/27/98
Support System Integration Test	02/27/98	03/27/98
Thor Development Complete	N/A	03/27/98

1.8.2 Dependencies

No.	Dependency Area	Dependency	Need Date
1			
2			
3			
4			

1.9 SYSTEM INTEGRITY THREAD SIMULATION REQUIREMENTS

No special Simulation Requirements.

1.10 SYSTEM INTEGRITY THREAD INTEGRATION AND SYSTEM TEST

1. The SCT will be modified by Activity Management to reflect various configurations of the Test Set.
   - SDE 1 & 2
   - IDE 1 & 2
2. Each Test Set will be loaded by Ops CM/Activity Management using the configurations defined above.
3. As the Test Set is loaded the System Status Viewer will monitor the activities to demonstrate the ability to load the system correctly and that the System Status Viewer can track the various states of each of the subsystems in the Test Set.
4. The System Status Viewer will be cycled through the detailed status of each of the Thor delivered Subsystems to demonstrate the availability and correct display of Subsystem Health Counter and Subsystem Status FD Information.
5. Each Subsystem will be forced to fail one by one while viewing the System Status Viewer to demonstrate that all elements of the thread track the failure of the Subsystems.
6. Bring new subsystems in to replace failed subsystems while viewing with the System Status Viewer to demonstrate that all elements of the thread track the introduction of new Subsystems to replace failed Subsystems.

1.11 SYSTEM INTEGRITY THREAD TRAINING REQUIREMENTS

1.11.1 Training Needed

Patrol

1.11.2 Training to be provided

None.

1.12 SYSTEM INTEGRITY THREAD FACILITIES REQUIREMENTS

None.

1.13 SYSTEM INTEGRITY THREAD ISSUES, ACTION ITEMS/RESOLUTION

1.13.1 Issues

1. Host Names Table — The issue is how the Host Names Table is built, controlled, and used.
   Thor Position — Host Names Tables will be created for each set, released through CM, and loaded by Ops CM.
   Long Term Position — Open
2. Location of System Integrity — The issue is where will System Integrity run (CCP vs DDP)
   Thor Position — System Integrity will run in the Master CCP.

Long Term Position — Open

3. Gateway Pseudos — How do Gateways access and set the equivalent of CCMS Buffer Access Table Pseudo FDs. The requirement is that LDB, PCM and possibly GSE Gateways have to have the capability to read Pseudo FDs and in some cases set them.
4. Gateway System Integrity Tables — How do Gateways get their System Status FD definitions.
   - Option 1 — OLDB is provided to the Gateway. Gateway Subsystem Init searches for FDs and establishes link.
   - Option 2 — Test Build builds a table for Gateways. Gateway Subsystem Init searches for FDs and establishes link.
5. Redundant Subsystems — What subsystems will be designed and implemented to be redundant?
6. Redundant Switchover — What is the real rate that Redundant Switchover is required.
7. User Application I/F to Subsystem Integrity — What is the requirements and design for the User Application Interface to Subsystem Integrity.
8. User Application Redundancy Management — Define the redundancy management concept for User Applications in CLCS.
   - Issue 1 — Fail Op or Fail Safe
   - Issue 2 — Degree of support from System to Support User Application Redundancy.

1.13.2 Action Items

1. Learn more about Service Processors & determine how they can be used.
2. Review COTS product (Patrol) for use in some of the processors.
3. Include RM & Network Services in White Paper.
4. Include Checkpoint Restart and Refresh concepts in System Integrity White Paper.
5. Analyze Gateway ability to determine health at update rate.
6. Get with SDC (Larry Carr) to determine what COTS product they are using for SSI/SI.

2. CSCI ASSESSMENTS

2.1 SYSTEM INTEGRITY ASSESSMENT

System Integrity executes in redundant Command and Control Processors (CCPs). This software performs the following functions:

1. System Integrity receives notification from each platform's Subsystem Integrity when each Subsystem has been Initialized and is Communicating.
2. System Integrity retrieves a health counter from each Subsystem Integrity at either the System Synchronous Rate or Display Synchronous Rate, depending on platform type.
3. System Integrity receives Subsystem Status Data from each platform's Subsystem Integrity at the Reporting Rate (TBD) including Use and Error Counters and other information similar to the System Device Matrix below.
4. System Integrity collects Status Updates and performance data provided by each Subsystem's Integrity on software executing in that Subsystem..
5. System Integrity incorporates information into health information from each Subsystem's Integrity on their capability to support applications.
6. System Integrity updates the master SCT and issues this information to the Subsystem Integrity on all platforms.
7. System Integrity will log and display to the workstations any status changes occurring in any platform.
8. System Integrity will provide Subsystem Status Data to the workstations upon demand, updating the data once a second as long as demanded.

System Device Matrix

System Device	Use Counter	Throughput	Error Counter	Status	Performance
DASD	Y	Y	Y	Y	Y
RTCN 1	Y	Y	Y	Y	Y
RTCN 2	Y	Y	Y	Y	Y
DCN 1	Y	Y	Y	Y	Y
DCN 2	Y	Y	Y	Y	Y
MP-1	N	N	Y	Y	Y
MP-N	N	N	Y	Y	Y

System Integrity Work Required

This is a list of work to be accomplished for this function.

System Integrity Assessment

CSC Name	CSC Labor (LM)	% of CSC
System Integrity	14 LM	60%

Basis of estimate

The overall amount of code in System Integrity is 2500 lines of C++ Code. It is estimated that approximately 60% of this will be accomplished in Thor. The code that won't be available in Thor is any that relates to Application Software Status or Redundancy Management. This will be incorporated in either Atlas or Titan.

Documentation

The following documentation will be produced for System Integrity.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	30
Users Guide	New	10
API Interface Document	New	5
Interface Design Document	New	5
Test Procedure	New	5

Assumptions

Open Issues

COTS tools, such as BMC's Patrol, should be evaluated to determine how much of System Integrity they could manage. Possible applications include handling Subsystem Status Data and/or software status.

2.2 COMMAND & CONTROL WORKSTATION SUBSYSTEM INTEGRITY ASSESSMENT

C&C Workstation (C&C WS) Subsystem Integrity executes in each Command and Control Workstation. This software performs the following functions:

Command and Control Workstation Subsystem Integrity Work Required

1. C&C WS Subsystem Integrity monitors the health and status of the Critical and Non-critical Processes in the C&C WS Subsystem and maintains a set of data that describe the health and status of the subsystem.
2. A set of System Status FDs will be created for all devices in the C&C WS
3. C&C WS Subsystem Integrity receives Status Updates for application software via Applications Services that defines the state of the applications running on the C&C WS.
4. C&C WS Subsystem Integrity will use the OLDB to determine what FD IDs to assign to the health and status data.
5. C&C WS Subsystem Integrity increments its health counter FD at the SSR, introduces it into the measurement data stream as long as all C&C WS Subsystem hardware and Critical Process Software are properly performing their function.
   - No subsystem errors are detected which are deemed terminal
   - No subsystem non-terminal errors are detected which degrade C&C WS subsystem support to a point that is deemed unacceptable
6. C&C WS Subsystem Integrity notifies System Integrity when the C&C WS Subsystem has been Initialized and is Communicating.
7. C&C WS Subsystem Integrity retrieves Use and Error Counters and other information similar to the data in the C&C WS System Device Matrix below and merges them as Subsystem Status FDs into the data stream at the defined recording rate.
8. C&C WS Subsystem Integrity determines the subsystem's capability to support applications and incorporates this information into the C&C WS Health FD.
9. C&C WS Subsystem Integrity calculates performance data based on data provided by other software executing in the C&C WS.
10. C&C WS Subsystem Integrity receives updates from System Integrity which updates the local copy of the SCT.
11. C&C WS Subsystem Integrity will notify System Integrity of major subsystem event changes using System Event Codes (SECs) as they occur.
12. C&C WS viewer will display, upon demand, any status changes received from System Integrity. The display will be updated at once a second as long as demanded.

Command and Control Workstation System Device Matrix

System Device	Use Counter	Throughput	Error Counter	Status	Performance
DASD	Y	Y	Y	Y	Y
RTCN 1	Y	Y	Y	Y	Y
RTCN 2	Y	Y	Y	Y	Y
DCN 1	Y	Y	Y	Y	Y
MP-1	N	N	Y	Y	Y
MP-N	N	N	Y	Y	Y

C&C W/S Subsystem Initialization Work Required

1. C&C W/S Subsystem Initialization will be modified to determine the FD IDs for the Health and Status data maintained by the various CSCs in the C&C W/S Subsystem and provide the information to C&C W/S Subsystem Integrity.
2. C&C W/S Subsystem Initialization will be modified to access the SCT and initialize gateway processes with data contained within it.

Command and Control Workstation CSCs Assessment

CSC Name	CSC Labor (LM)	% of CSC
C&C WS Subsystem Integrity	3 LM	60%
C&C WS Subsystem Initialization	1 LM	30%

Basis of estimate

The overall amount of code in C&C WS Subsystem Integrity is 750 lines of C++ Code. It is estimated that approximately 60% of this will be accomplished in Thor. The code that won't be available in Thor is any that relates to Application Software Status or Redundancy Management. This will be incorporated in either Atlas or Titan.

Documentation

The following documentation will be produced for C&C WS Subsystem Integrity.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	30
Users Guide	New	10
API Interface Document	New	5
Interface Design Document	New	5
Test Procedure	New	5

Assumptions

None

Open Issues

None

2.3 CCP SUBSYSTEM INTEGRITY ASSESSMENT

CCP Subsystem Integrity executes in a Command and Control Processor. This software performs the following functions:

CCP Subsystem Integrity Work Required

1. CCP Subsystem Integrity monitors the health and status of the Critical and Non-critical Processes in the CCP Subsystem and maintains a set of data that describe the health and status of the subsystem.
2. A set of System Status FDs will be created for all devices in the CCP.
3. CCP Subsystem Integrity receives Status Updates for application software via Applications Services that defines the state of the applications running on the CCP.
4. CCP Subsystem Integrity will use the OLDB to determine what FD IDs to assign to the health and status data.
5. CCP Subsystem Integrity increments its health counter FD at the SSR, introduces it into the measurement data stream as long as all CCP Subsystem hardware and Critical Process Software are properly performing their function.
   - No subsystem errors are detected which are deemed terminal
   - No subsystem non-terminal errors are detected which degrade CCP subsystem support to a point that is deemed unacceptable
6. CCP Subsystem Integrity notifies System Integrity when the CCP Subsystem has been Initialized and is Communicating.
7. CCP Subsystem Integrity Receives Status Updates for application software via Applications Services that defines the state of the applications running on the CCP.
8. CCP Subsystem Integrity sends a health counter to System Integrity at the System Synchronous Rate.
9. CCP Subsystem Integrity retrieves Use and Error Counters and other information similar to the information contained in the CCP System Device Matrix below and merges them as Subsystem Status FDs into the data stream at the defined recording rate.
10. CCP Subsystem Integrity determines the CCP Subsystem's Capability to support and incorporates this information into Subsystem Health FD for System Integrity.
11. CCP Subsystem Integrity calculates performance data based on data provided by other software executing in the CCP.

12. CCP Subsystem Integrity will notify System Integrity of major subsystem event changes using System Event Codes (SECs) as they occur.
13. CCP Subsystem Integrity receives updates from System Integrity and updates the local copy of the SCT.

CCP System Device Matrix

System Device	Use Counter	Throughput	Error Counter	Status	Performance
DASD	Y	Y	Y	Y	Y
RTCN 1	Y	Y	Y	Y	Y
RTCN 2	Y	Y	Y	Y	Y
DCN 1	Y	Y	Y	Y	Y
MP-1	N	N	Y	Y	Y
MP-N	N	N	Y	Y	Y

CCP Subsystem Initialization Work Required

1. CCP Subsystem Initialization will be modified to determine the FD IDs for the Health and Status data maintained by the various CSCs in the CCP Subsystem and provide the information to CCP Subsystem Integrity.
2. CCP Subsystem Initialization will be modified to access the SCT and initialize gateway processes with data contained within it.

CCP Subsystem CSCs Assessment

CSC Name	CSC Labor (LM)	% of CSC
CCP Subsystem Integrity	3.0	60%
CCP Subsystem Init	1.0	30%

Basis of estimate

The overall amount of code in CCP Subsystem Integrity is 750 lines of C++ Code. It is estimated that approximately 60% of this will be accomplished in Thor. The code that won't be available in Thor is any that relates to redundancy Management. This will be incorporated in later releases. Subsystem Initialization code is estimated to be 250 lines of low complexity C Code.

Documentation

The following documentation will be produced for CCP Subsystem Integrity.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	30
Users Guide	New	10
API Interface Document	New	5
Interface Design Document	New	5
Test Procedure	New	5

Assumptions

None

Open Issues

None

2.4 DDP SUBSYSTEM INTEGRITY ASSESSMENT

DDP Subsystem Integrity executes in a Data Distribution Processor. This software performs the following functions:

DDP Subsystem Integrity Work Required

1. DDP Subsystem Integrity monitors the health and status of the Critical and Non-critical Processes in the DDP Subsystem and maintains a set of data that describe the health and status of the subsystem.
2. A set of System Status FDs will be created for all devices in the DDP.
3. DDP Subsystem Integrity receives Status Updates for application software via Applications Services that defines the state of the applications running on the DDP.
4. DDP Subsystem Integrity will use the OLDB to determine what FD IDs to assign to the health and status data.
5. DDP Subsystem Integrity increments its health counter FD at the SSR, introduces it into the measurement data stream as long as all DDP Subsystem hardware and Critical Process Software are properly performing their function.
   - Data Distribution
   - Data Health
   - Data Fusion
   - No subsystem errors are detected which are deemed terminal
   - No subsystem non-terminal errors are detected which degrade C&C WS subsystem support to a point that is deemed unacceptable
6. DDP Subsystem Integrity notifies System Integrity when the DDP Subsystem has been Initialized and is Communicating.
7. DDP Subsystem Integrity Receives Status Updates for application software via Applications Services that defines the state of the applications running on the DDP.
8. DDP Subsystem Integrity determines the DDP Subsystem's Capability to support and incorporates this information into it's Subsystem Health FD at the System Synchronous Rate.
9. DDP Subsystem Integrity retrieves Use and Error Counters and other information similar to the data in the DDP System Device Matrix below and merges them as Subsystem Status FDs into the data stream at the defined recording rate.
10. DDP Subsystem Integrity calculates performance data based on data provided by other software executing in the DDP.
11. DDP Subsystem Integrity will notify System Integrity of major subsystem event changes using System Event Codes (SECs) as they occur.
12. DDP Subsystem Integrity receives updates from System Integrity and updates the local copy of the SCT.

DDP System Device Matrix

System Device	Use Counter	Throughput	Error Counter	Status	Performance
DASD	Y	Y	Y	Y	Y
RTCN 1	Y	Y	Y	Y	Y
RTCN 2	Y	Y	Y	Y	Y
DCN 1	Y	Y	Y	Y	Y
MP-1	N	N	Y	Y	Y
MP-N	N	N	Y	Y	Y

DDP Subsystem Initialization Work Required

1. DDP Subsystem Initialization will be modified to determine the FD IDs for the Health and Status data maintained by the various CSCs in the DDP Subsystem and provide the information to DDP ubsystem Integrity.
2. DDP Subsystem Initialization will be modified to access the SCT and initialize gateway processes with data contained within it.

DDP Subsystem Integrity Assessment

CSC Name	CSC Labor (LM)	% of CSC
DDP Subsystem Integrity	3.0	60%
DDP Subsystem Init	1.0	30%

Basis of estimate

The overall amount of code in DDP Subsystem Integrity is 750 lines of C++ Code. It is estimated that approximately 60% of this will be accomplished in Thor. The code that won't be available in Thor is any that relates to Redundancy Management. This will be incorporated in later releases. Subsystem Initialization code is estimated to be 250 lines of low complexity C Code.

Documentation

The following documentation will be produced for DDP Subsystem Integrity.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	30
Users Guide	New	10
API Interface Document	New	5
Interface Design Document	New	5
Test Procedure	New	5

Assumptions

None

Open Issues

None

2.5 DATA DISTRIBUTION AND PROCESSING ASSESSMENT

Changes to support Thor System Integrity are covered in the Data Distribution and Processing Thread.

2.6 GSE GATEWAY SUBSYSTEM INTEGRITY ASSESSMENT

A new CSC (GSE Subsystem Integrity) will be created and integrated into the GSE Gateway to support the System Integrity Thread.

GSE Subsystem Integrity Work Required

1. GSE Subsystem Integrity monitors the health and status of the Critical and Non-critical Processes in the GSE Subsystem and maintains a set of data that describe the health and status of the subsystem.
2. A set of System Status FDs will be created for all devices in the GSE Gateway
3. GSE GW Subsystem Integrity will use the OLDB to determine what FD IDs to assign to the health and status data.
4. GSE Subsystem Integrity increments its health counter FD at the SSR, introduces it into the measurement data stream, and sends it to the DDP as part of the Change Data Packet Payload as long as all GSE Subsystem hardware and software are properly performing their function.
   - Bus polling and data processing is operational and working properly
   - No subsystem errors are detected which are deemed terminal
   - No subsystem non-terminal errors are detected which degrade GSE subsystem support to a point that is deemed unacceptable
5. The System Status FDs will be sent to the DDP as part of the Change Data Packet Payload once per second.
6. GSE Subsystem Integrity calculates performance data based on data provided by other software executing in

the GSE Gateway.
7. GSE Subsystem Integrity will notify System Integrity of major subsystem event changes using System Event Codes (SECs) as they occur.
8. GSE Subsystem Integrity updates its SCT upon receipt of an SEC from System Integrity that relates to the status of another platform in the Set.

GSE Gateway Initialization Work Required

1. GSE Gateway Initialization will be modified to determine the FD IDs for the Health and Status data maintained by the various CSCs in the GSE Gateway and provide the information to GSE Subsystem Integrity.
2. GSE Gateway Initialization will be modified to access the SCT and initialize gateway processes with data contained within it.

GSE Bus Polling and Data Processing Work Required

1. GSE Bus Polling and Data Processing will maintain health and use counters for the GSE Data Bus.
2. GSE Bus Polling and Data Processing provides the health and use counters to GSE Subsystem Integrity.
3. GSE Bus Polling and Data Processing maintains the status of all HIMs attached to the bus (Polling Active/Inhibited, Test Active/Inhibited)
4. GSE Bus Polling and Data Processing maintains the status of the following information related to the GSE Data Bus and provides them to GSE Subsystem Integrity:
   - Interface started
   - Measurement Data Valid
   - Table Load Error
   - Tables Loaded
   - Commands Enabled
   - Processing Enabled
   - T/R 1
   - T/R 2
   - Global HIM Testing Enabled

CSC Name	CSC Labor (LM)	% of CSC
GSE Subsystem Integrity	6 LM	60%
GSE Gateway Initialization	1 LM	10%
GSE Polling and Data Processing	2 LM	N/A

Basis of estimate

GSE Subsystem Integrity is estimated to be approximately 500 lines of medium to high complexity code. Approximately 60% of the code will be developed for Thor. The amount of code to perform this function in the GSE Gateway is approximately 200 Lines of low complexity code. The amount of code change in GSE Polling and Data Processing is approximately 400 lines of low to medium complexity code.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	15
Users Guide	N/A	N/A
API Interface Document	New	5
Interface Design Document	N/A	N/A
Test Procedure	N/A	N/A

Assumptions

None

Open Issues

None

2.7 PCM GATEWAY SUBSYSTEM INTEGRITY ASSESSMENT

A new CSC (PCM Subsystem Integrity) will be created and integrated into the PCM Gateway to support the System Integrity Thread.

PCM Subsystem Integrity Work Required

1. PCM Subsystem Integrity monitors the health and status of the Critical and Non-critical Processes in the PCM Subsystem and maintains a set of data that describe the health and status of the subsystem.
2. A set of System Status FDs will be created for all devices in the PCM Gateway
3. PCM GW Subsystem Integrity will use the OLDB to determine what FD IDs to assign to the health and status data.
4. PCM Subsystem Integrity increments its health counter FD at the SSR, introduces it into the measurement data stream, and sends it to the DDP as part of the Change Data Packet Payload as long as all PCM Subsystem hardware and software are properly performing their function.
   - PCM Measurement Processing is operational and working properly
   - No subsystem errors are detected which are deemed terminal
   - No subsystem non-terminal errors are detected which degrade PCM subsystem support to a point that is deemed unacceptable
5. The System Status FDs will be sent to the DDP as part of the Change Data Packet Payload once per second.
6. PCM Subsystem Integrity calculates performance data based on data provided by other software executing in the PCM Gateway.
7. PCM Subsystem Integrity will notify System Integrity of major subsystem event changes using System Event Codes (SECs) as they occur.
8. PCM Subsystem Integrity updates its SCT upon receipt of an SEC from System Integrity that relates to the status of another platform in the Set.

PCM Gateway Initialization Work Required

1. PCM Gateway Initialization will be modified to determine the FD IDs for the Health and Status data maintained by the various CSCs in the PCM Gateway and provide the information to PCM Subsystem Integrity.
2. PCM Gateway Initialization will be modified to access the SCT and initialize gateway processes with data contained within it.

Other PCM CSCs Work Required

1. PCM Gateway CSCs will maintain health and use counters for the PCM Gateway.
2. PCM Gateway CSCs provide the health and use counters to PCM Subsystem Integrity.
3. PCM Gateway CSCs maintain the status of all the PCM equipment and adapters (e.g., Measurement Processing Active/Inhibited, Test Active/Inhibited)
4. The PCM Gateway maintains the status of the following information related to the PCM Gateway and provides them to PCM Subsystem Integrity: For example data like the following will be provided:
   - Interface started
   - Measurement Data Valid
   - Table Load Error
   - Tables Loaded
   - Processing Enabled
   - Additional Indicators available from the PCM Interface

CSC Name	CSC Labor (LM)	% of CSC
PCM Subsystem Integrity	6 LM	60%
PCM Gateway Initialization	1 LM	10%
Other PCM Gateway CSCs	2 LM	N/A

Basis of estimate

PCM Subsystem Integrity is estimated to be approximately 500 lines of medium to high complexity code. Approximately 60% of the code will be developed for Thor. The amount of code to perform this function in the PCM Gateway is approximately 200 Lines of low complexity code. The amount of code change in PCM Measurement Processing is approximately 400 lines of low to medium complexity code.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	15
Users Guide	N/A	N/A
API Interface Document	New	5
Interface Design Document	N/A	N/A
Test Procedure	N/A	N/A

Assumptions

None

Open Issues

None

2.8 LDB GATEWAY SUBSYSTEM INTEGRITY ASSESSMENT

A new CSC (LDB Subsystem Integrity) will be created and integrated into the LDB Gateway to support the System Integrity Thread.

LDB Subsystem Integrity Work Required

1. LDB Subsystem Integrity monitors the health and status of the Critical and Non-critical Processes in the LDB Subsystem and maintains a set of data that describe the health and status of the subsystem.
2. A set of System Status FDs will be created for all devices in the LDB Gateway
3. LDB GW Subsystem Integrity will use the OLDB to determine what FD IDs to assign to the health and status data.
4. LDB Subsystem Integrity increments its health counter FD at the TBD rate, introduces it into the measurement data stream, and sends it to the DDP as part of a Change Data Packet Payload as long as all LDB Subsystem hardware and software are properly performing their function.
   - Bus polling and data processing is operational and working properly
   - No subsystem errors are detected which are deemed terminal
   - No subsystem non-terminal errors are detected which degrade LDB subsystem support to a point that is deemed unacceptable
5. The System Status FDs will be sent to the DDP as part of a Change Data Packet Payload once per second.
6. LDB Subsystem Integrity calculates performance data based on data provided by other software executing in the LDB.
7. LDB Subsystem Integrity will notify System Integrity of major subsystem event changes using System Event Codes (SECs) as they occur.

8. LDB Subsystem Integrity updates its SCT upon receipt of an SEC from System Integrity that relates to the status of another platform in the Set.

LDB Gateway Initialization Work Required

1. LDB Gateway Initialization will be modified to determine the FD IDs for the Health and Status data maintained by the various CSCs in the LDB Gateway and provide the information to LDB Subsystem Integrity.
2. LDB Gateway Initialization will be modified to access the SCT and initialize gateway process with data contained within it.

LDB Process CCP Request Word Required

1. LDB Process CCP Request will maintain health counters and provide this information to LDB Subsystem Integrity.

2. LDB Process CCP Request maintains the status of the following information related to the LDB Data Bus and provides them to LDB Subsystem Integrity:
   - Commands Status (i.e., Enabled/Disabled)
   - Table Load Error
   - Tables Loaded

LDB Bus Interface Control Work Required

1. LDB Bus Interface Control will monitor LDB Bus Interface hardware operation and status and make provide this information to LDB Bus Communications Processing.

LDB Bus Communications Processing Work Required

1. LDB Bus Communications Processing will maintain health and use counters for the LDB Data Bus based on data obtained from LDB Bus Interface Control..
2. LDB Bus Communications Processing provides the health and use counters to LDB Subsystem Integrity.
3. LDB Bus Communications Processing maintains the status of the following information related to the LDB Data Bus and provides them to LDB Subsystem Integrity: For example, information such as the following will be maintained by the LDB Gateway:
   - Interface started
   - Mode of Operation
   - I/F Initialization Status (i.e., Started/Stopped)
   - T/R1 Status
   - T/R2 Status
   - Bus Error Counts
   - Bus Switch Counts

CSC Name	CSC Labor (LM)	% of CSC
LDB Subsystem Integrity	6 LM	60%
LDB Process CCP Request	1 LM	5%
LDB Bus Interface Control	1 LM	5%
LDB Bus Communications Processing	1 LM	5%

Basis of estimate

LDB Subsystem Integrity is estimated to be approximately 500 lines of medium to high complexity code. Approximately 60% of the code will be developed for Thor. The amount of code to perform this function in the LDB Gateway is approximately 200 Lines of low complexity code. The amount of code change in LDB Process CCP Request is approximately 200 lines of low to medium complexity code. The amount of code change in LDB

Bus Interface Control is approximately 200 lines of low to medium complexity code. The amount of code change in LDB Bus Communications Processing is approximately 200 lines of low to medium complexity code.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	15
Users Guide	N/A	N/A
API Interface Document	New	5
Interface Design Document	N/A	N/A
Test Procedure	N/A	N/A

Assumptions

None

Open Issues

None

2.9 SYSTEM STATUS VIEWER ASSESSMENT

CSC System Status Viewer

1. System Status Viewer provides a system status HCI display that shows overall Test Set status of all subsystems.
2. System Status Viewer provides a summary level display which will show status of all subsystems on a single page.
3. System Status Viewer provides a single subsystem display for the following subsystem delivered in Thor:
    - GSE Gateway
    - PCM D/L Gateway
    - LDB Gateway
    - Consolidated Gateway
    - CCP
    - DDP
    - C&C W/S
4. Each subsystem display will show details of information collected by SSI in each of the above subsystems.
5. System Status Viewer displays will be refreshed once per second.
6. System Status Viewer will baseline system messages using the System Message Catalog. These messages will contain message and help text.

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
System Status Viewer	6	70%

Basis of estimate

CSC System Status Viewer may use a code generator. The overall amount of code in Subsystem Integrity is 10,000 lines of C++ and X/motif or JAVA. The decision of which to use will be made at DP2.

Documentation

Provide your assessment of the kinds and amount of documentation that must be provided with the capability.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New	30
Users Guide	New	15
Interface Document	New	5
Test Procedure	New	5

Assumptions

N/A

Open Issues

N/A

2.10 APPLICATION SERVICES ASSESSMENT

The following work needs to be done for the Thor delivery:

- Provide a C++ API for us on the non-Gateway systems that in effect will tell each requester "who am I". [Gateway systems will use a C API provided by Systems Services.] Typically this information will consist of all known aliases for the program (physical and logical) extracted from the System Configuration Table.
- Provide an API for applications to report in (give health to) Subsystem Integrity.

CSC TBD Work Required

ASV CSC TBD will handle both APIs mentioned above.

CSC Name	CSC Labor (LM)	% of CSC
ASV CSC TBD	3.0	100%

Basis of estimate

No re-use code is available for these APIs. The work includes design, system design support, analysis, coding, testing, and documentation.

Documentation

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	New DP2 & DP3	12
Users Guide	N/A	
API Interface Document	API data provided in web format	10 web pages
Interface Design Document	N/A	
Test Procedure	UT, UIT, CIT	30

Assumptions

System Services will provide the C interface for use by the Gateways. Identification information will be available in the SCT. SCT data is available through API to System Services. CIT testing will show that Applications Services and underlying System Service provide correct "who I am" data.

Health information will be defined by the thread.

Open Issues

No large issues.
Need definition of health information.

2.11 DBSAFE AND TEST BUILD ASSESSMENT

DBSAFE and Test Build will be modified to allow System Status FDs to be defined for all Subsystem Integrity FDs. These FDs will be included in the On Line Data Bank for use by programs which need them.

DBSAFE Work Required

The DBSAFE baseline code already provides for support for System Integrity using SSA1 and SSA2 type function designators. However, since this baseline was copied over from the CCMS DBSAFE, there will be some modifications necessary to support this capability. These modifications will be mainly to delete fields carried from CCMS which are not required by CLCS.

Test Build and Control Assessment

Include support in the TCID for System Integrity using SSA1 and SSA2 type function designators. This work will allow for required changes to Oracle tables and code to support the inclusion of SSA1 and SSA2 FDs in the FD Directory and the views to support OLDB, Simulation, and TDRR.

CSC Name	CSC Labor (LM)	% of CSC
DBSAFE	2 LM	N/A
Test Build	2 LM	N/A

Basis of estimate

The DBSAFE estimate covers the required Oracle Table changes as required to support this capability. The Test Build estimate covers the required changes required to support this capability.

Documentation

The following documentation will be updated as required.

Example:

Document Type	New/Update	Number of Pages
DBSAFE Requirements and Design Documentation	Update	TBD
Test Build Requirements and Design Documentation	N/A	TBD
DBSAFE & Test Build Test Procedure	TBD	TBD

Assumptions

It is assumed that the SSA1 and SSA2 FD types will support all functions needed for this capability.

Open Issues

None.

3. HWCI ASSESSMENTS

There are no HWCIs that are part of this thread.

4. COTS PRODUCTS DEPENDENCIES

4.1 SW PRODUCTS DEPENDENCY LIST

None.

4.2 HW PRODUCTS DEPENDENCY LIST

None.

5. ATTACHMENTS

5.1 SYSTEM CONFIGURATION TABLE

The following information is being developed in support of the System Integrity Thread. Some of it may be developed in Thor and others will certainly be developed at a later date.

1. There is a global SCT on the SDC that contains information about every Ref Des that exists anywhere.
2. This global SCT may be tailored by the Activity Management function (executing in a office workstation or on the Set Master C&C WS to reflect the Test Set Configuration.
3. The Test Set Master can further tailor the SCT for the operations currently taking place in the Test Set.

Location	Test Set Name or ID								
Ref Des	Host Name	Subsystem Name	Subsystem Type	Attached to	Current State				

Definitions:

Location: LCC, HMF, CITE, SAIL, Dryden, SDE 1, SDE 2, IDE 1. IDE 2

Ref Des: Reference Designation of the machine.

Host Name: UNIX Host Name of the machine.

Subsystem Name: GSE 1-5 {A/S}, SSME {A/S}, OIGPC {A/S}, LDB {A/S}, CCP1-8 {A/S}, DDP1-2 {A/S}, C&C 1-48, OCM

Subsystem Type: GSE, ME PCM, OI PCM, LDB, CCP, DDP, C&C WS, OCM

Attached to: BUS IDs, PCM Streams, ????

Current State:
- In Configuration T/F
- TCID Loaded T/F
- Communicating T/F
- Go T/F
- Part of Redundant Pair/Triplet

Set Master C&C W/S The workstation used in a configurable set (e.g., LCC) to monitor and control Test Sets and to allocate resources within the within the Set's domain.

Test Set Master C&C W/S The workstation used in a Test Set to control resources within the Test Set's domain.

5.2 SYSTEM STATE MATRIX

SYSTEM STATES	VISIBLE TO TC	SYSTEM CAPABILITIES IN THIS STATE	ACCEPTABLE COMMANDS	CMD CAUSING TRANSITION NEXT STATE	RESULT OF COMMAND	RETURN TO STATE COMMAND
In Config	X	None	Power On			
				Power On Boot	OS Loaded & Initialized (GW Init SCID)	
Platform Init'd (GW - N/A)		UNIX Based Comm	Init SCID			Power Off
				Init SCID	SCID Initialized & Limited Comm	
SCID Initialized		RTPS Comm (Limited)	Init SCID or TCID			Shutdown
				Init TCID	TCID Initialized	
Loaded	X	RTPS Comm (Limited)	Activate Cmd			Init SCID, In TCID, Shutd
				Activate	HC Started & Full Comm	
Comm	X	Health Counts, Full Comm	Nearly Full Comm, No End Item EI Cmds, MDTM			Terminate, Shutdown
				A DA	Starts Processing, Data Acquisition	
GO Operational (Active)	X	Data Acq, Full Comm, SSI Poll &Xmit Data Changes.	All except Init SCID, TCID, Activate			Terminate, Shutdown, I
GO Operational (Standby)	X	Data Acq, Full Comm, SSI, Monitor Poll, No Xmit	All except Init SCID, TCID, Activate			Terminate, Shutdown, I
ORT	X	Full comm., Diagnostics			Begins Operational Readiness Test	Exit ORT, Terminate, Shutdown